# 머신러닝 & 딥러닝 5

AI 학술동아리 <MLP>

# - Index

1. 비지도 학습 - 군집
2. k-평균 알고리즘(최적의 k값 찾기)
3. 차원, 차원 축소(PCA), 설명된 분산

# Classical Machine Learning

Task Driven

Data Driven

## Supervised Learning

( Pre Categorized Data )

### Classification

( Divide the socks by Color )

Eg. Identity Fraud Detection

### Regression

( Divide the Ties by Length )

Eg. Market Forecasting

## Unsupervised Learning

( Unlabelled Data )

### Clustering

( Divide by Similarity )

Eg. Targeted Marketing

### Association

( Identify Sequences )

Eg. Customer Recommendation

### Dimensionality Reduction

( Wider Dependencies )

Eg. Big Data Visualization

Obj: Predications & Predictive Models

Pattern/ Structure Recognition

# CLASSICAL MACHINE LEARNING

Data is pre-categorized or numerical

## SUPERVISED

Predict a category

### CLASSIFICATION

«Divide the socks by color»

Predict a number

### REGRESSION

«Divide the ties by length»

Data is not labeled in any way

## UNSUPERVISED

Divide by similarity

### CLUSTERING

«Split up similar clothing into stacks»

Find hidden dependencies

### DIMENSION REDUCTION (generalization)

«Make the best outfits from the given clothes»

Identify sequences

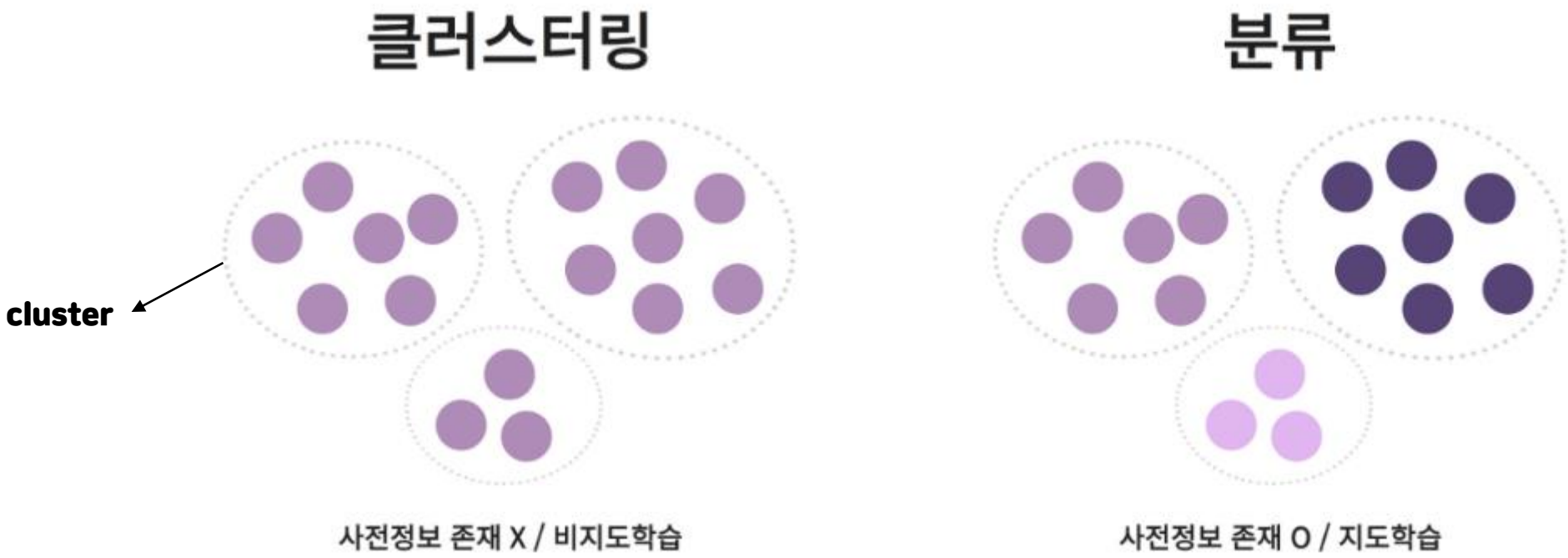
### ASSOCIATION

«Find what clothes I often wear together»

# 1. Unsupervised Learning(비지도 학습) - Clustering(군집)

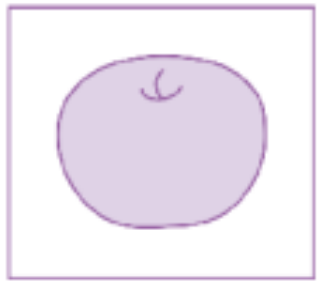
**Clustering(군집) : 비슷한 샘플끼리 그룹으로 모으는 작업**

**Clustering Algorithm에서 만든 그룹을 Cluster(클러스터)라고 부름**

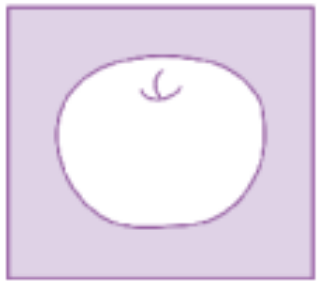


# Tip! 흑백 샘플 이미지

원본 흑백 이미지



색상 반전 이미지

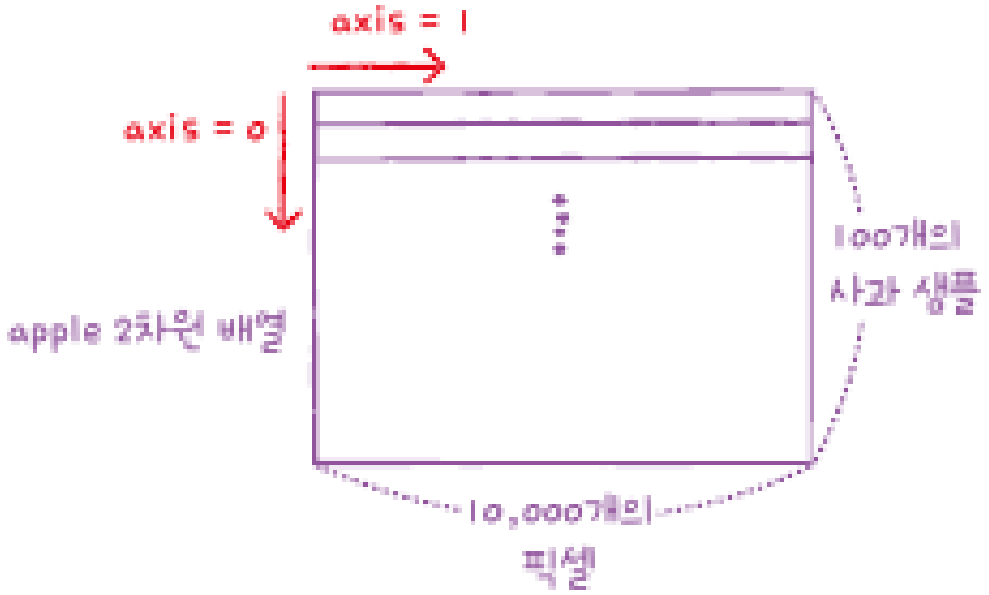


픽셀은 0~255의 값을 가짐
- 0에 가까울수록 검은색
- 255에 가까울수록 흰색

알고리즘이 어떤 출력을 만들기 위해 곱셈, 덧셈을 함
픽셀값이 0(흰색)이면 출력도 0이 되어 의미가 없음
-> 픽셀값이 높으면 출력값도 커지기 때문에 의미를 부여하기 좋음

=> 색상 반전을 시킴

# Tip! axis=0, axis=1

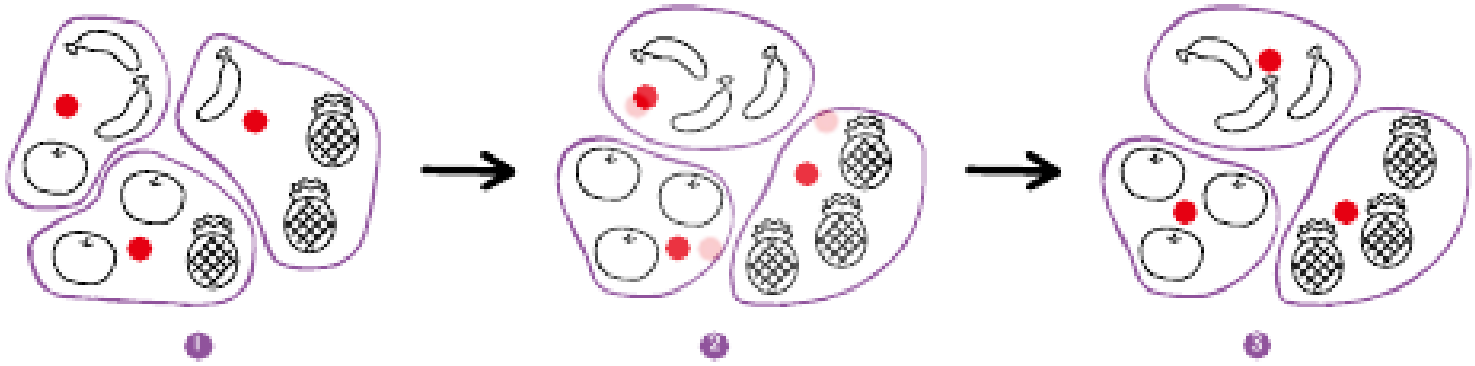



**apple.shape == (100, 10000)**

# 2. k-Means Algorithm

k-means algorithm이 각 군집의 평균값을 자동으로 찾아줌
평균값 - cluster의 중심에 위치하기 때문에,
cluster center(클러스터 중심) 또는 centroid(센트로이드)라고 불림

k-means algorithm 작동 방식
1. 무작위로 k개의 centroid를 정하기
2. 각 샘플에서 가장 가까운 centroid를 찾아 해당 cluster의 샘플로 지정
3. cluster에 속한 샘플의 평균값으로 centroid를 변경
4. centroid에 변화가 없을 때까지 2번으로 돌아가 반복



centroid를 특성 공학처럼 사용해 데이터셋을 저차원으로 변환 가능

feature
한 이미지 픽셀 수 -> 각 centroid까지의 거리
ex) 10,000 -> 3

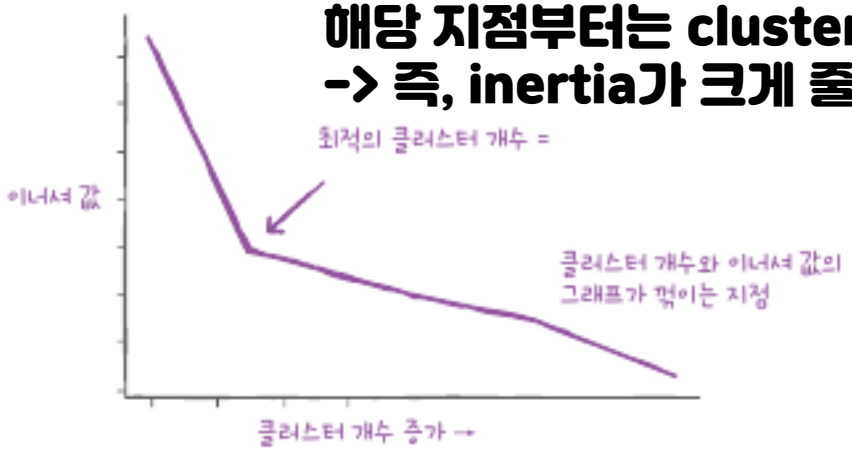
# 2-1. 최적의 k 찾기

실전에서는 몇 개의 cluster가 있는지 알 수 없음 -> 적절한 k값을 찾기 - elbow(엘보우) 방법

k-means 알고리즘은 centroid와 cluster에 속한 샘플 사이의 거리를 잴 수 있음,
거리의 제곱 합 = inertia(이너셔)
inertia - cluster에 속한 샘플이 얼마나 가깝게 모여 있는지를 나타내는 값
일반적으로 cluster의 개수가 늘어나면 cluster 개개의 크기는 줄어듦 -> inertia도 줄어듦

elbow 방법 : cluster의 개수를 늘려가며, inertia의 변화를 관찰하여 최적의 cluster 개수를 찾음

cluster의 개수를 증가시키면서 inertia를 그래프로 그리면 감소하는 속도가 꺾이는 지점이 존재
(해당 지점이 마치 팔꿈치 모양이라 elbow 방법)
해당 지점부터는 cluster의 개수를 늘려도 cluster에 잘 밀집된 정도가 크게 개선되지 않음
-> 즉, inertia가 크게 줄어들지 않음 => 해당 지점이 최적의 cluster 개수



# 3-1. Dimension(차원)

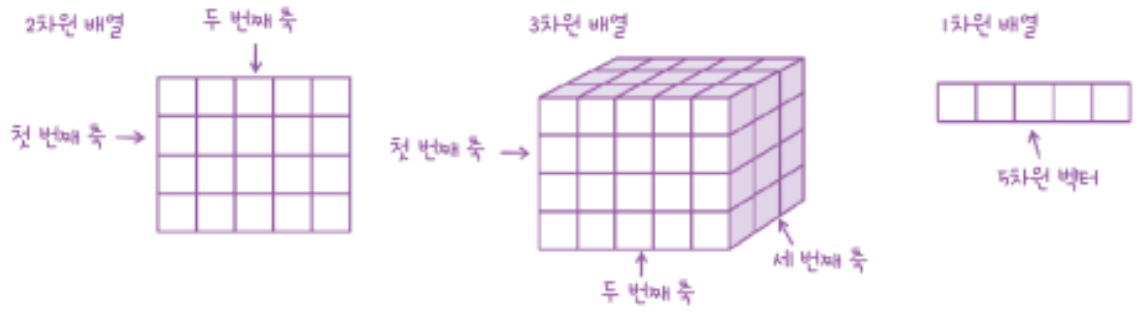
**데이터가 가진 속성 = feature = dimension**

**사진의 경우 픽셀 수가 feature(dimension) 수**

**dimension을 줄일 수 있다면 저장 공간을 크게 절약할 수 있음**

✚ 여기서 잠깐 **2차원 배열과 1차원 배열의 차원은 다른 건가요?**

2차원 배열과 1차원 배열(벡터)에서 차원이란 용어는 조금 다르게 사용합니다. 다차원 배열에서 차원은 배열의 축 개수가 됩니다. 가령 2차원 배열일 때는 행과 열이 차원이 되죠. 하지만 1차원 배열, 즉 벡터일 경우에는 원소의 개수를 말합니다. 다음 그림을 참고하세요.




이 절에서는 혼돈을 피하고자 가능하면 차원 대신 특성을 사용합니다. 하지만 차원이란 단어를 완전히 배제하기는 어렵습니다. 이 책이나 다른 책을 볼 때 참고하세요.

# 3-2. Dimensionality Reduction(차원 축소)

feature가 많으면 linear 모델의 성능이 높아지고 훈련 데이터에 쉽게 overfitting됨

dimensionality reduction은 데이터를 가장 잘 나타내는 일부 feature를 선택하여,
데이터 크기를 줄이고 지도 학습 모델의 성능을 향상시킬 수 있는 방법
=> 저장 공간 효율적으로 이용 가능, 머신러닝 모델 훈련 속도 높아짐

훈련 데이터의 dimension을 3개 이하로 줄이면 화면에 출력하기 쉬움 -> 비교적 시각화하기 쉬움

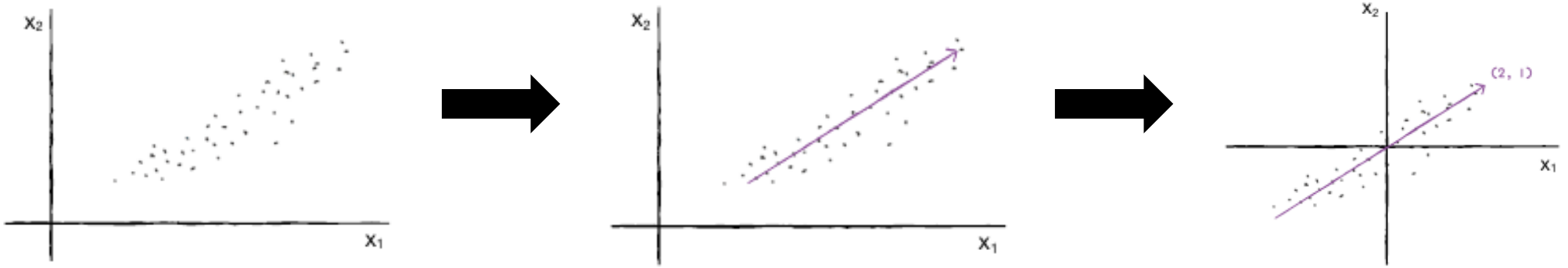
줄어든 dimension에서 다시 원본 dimension으로 손실을 최대한 줄이면서 복원 가능

대표적인 dimensionality reduction algorithm은
Principal Component Analysis(주성분 분석) = PCA

# 3-2-1. PCA

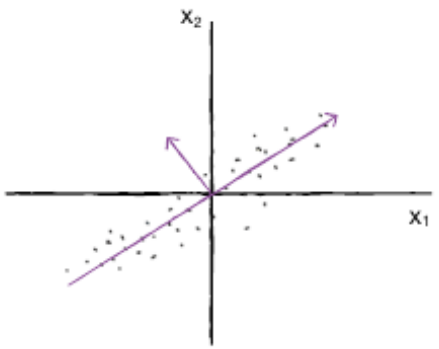
PCA - 데이터에 있는 **분산이 큰 방향**을 찾는 것 분산 : 데이터가 널리 퍼져있는 정도

-> **데이터(데이터 셋의 어떤 특징)를 잘 표현하는 어떤 벡터 : 주성분**





샘플 데이터 s(4, 2)를 주성분에 직각으로 투영하여
1차원 데이터 p(4, 5)를 만듦




기술적인 이유로 주성분은 원본 feature의 개수와 샘플 개수 중 작은 값만큼 찾을 수 있음
일반적으로 비지도 학습은 대량의 데이터에서 수행하기 때문에 원본 feature의 개수만큼 찾을 수 있다고 말함

**일반적으로 주성분은 원본 feature의 개수만큼 찾을 수 있음**

# 3-3. Explained Variance(설명된 분산)

explained variance : 주성분이 원본 데이터의 분산을 얼마나 잘 나타내는지 기록한 값

첫 번째 주성분의 분산이 가장 큼

모든 주성분의 분산 비율을 더하면 총 분산 비율을 얻을 수 있음

총 분산 비율이 높을수록, 축소한 데이터를 원본과 비슷하게 복원 가능